2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Monitoring autonomous persistent surveillance missions using invariance

Vladislav Nenchev and Prodromos Sotiriadis

Abstract— **This paper studies runtime monitoring for persistent surveillance by autonomous robots when the autonomy stack is a black box. The environment is partitioned into finitely many parts, each carrying an uncertainty state that decreases when observed and increases otherwise. We model the closed loop as a state-dependent hybrid system with linear parameter varying dynamics and design a monitor based on an invariant computed offline. As this invariant is typically hard to obtain for large to-be-surveyed spaces, we propose a compositional monitor obtained by decentralized computation of low-dimensional invariant sets for each uncertainty region, and checking their conjunction online. Under common independence assumptions, the compositional monitor is sound and complete with respect to the full-system invariant. The approach is applied in a case study with a real robot persistently monitoring a labyrinth, emphasizing its applicability in practice.**

I. Introduction

In persistent surveillance, an agent moves through an environment to collect information about specific targets over time. It is applicable across a wide range of robotic problems, such as ocean monitoring [1] and forest fire detection [2]. In many deployed autonomous systems, navigation stacks mix Artificial Intelligence (AI)-based modules with proprietary or otherwise black-box components. Unanticipated disturbances, sensor faults, and software vulnerabilities can trigger navigation stalls, collisions, or critical failures. In addition, cyber attacks that spoof sensors or manipulate control channels can deliberately degrade performance or subvert mission objectives. This motivates a runtime monitoring layer that reasons from measured state (and, when available, provided inputs) and raises early alerts before loss of progress manifests operationally.

This paper studies monitoring the operation of a robot performing persistent surveillance when the surveyed workspace is partitioned into a finite number of parts. We model the uncertainty state of each part as a Linear Parameter-Varying (LPV) system, as we assume that the robot has the ability to take (noisy) measurements of the state; this is a typical setting that can be found in many persistent surveillance approaches such as [3], [4], [5], [6]. The uncertainty *decreases* while remaining within the part and *increases* otherwise. We represent the closed loop as a state-dependent hybrid system characterized by Piece-Wise Affine (PWA) dynamics, incorporating bounded, region-dependent LPV uncertainty, and compute the Robust Controlled Invariant Set (RCIS) [7]. Intuitively, if the current measurement lies in a RCIS, then for all admissible inputs and parameter variations consistent with the current region, the next state also remains inside. This property of the RCIS is used for runtime monitoring.

Computing the maximal RCIS in the *full* space that contains the robot state, all uncertainties of the to-be-surveyed regions, and inputs becomes intractable as the number of parts of the environment grows. We propose a *compositional* construction: for each part we compute the RCIS in robot pose, input, and part-uncertainty only, and the online monitor checks the *conjunction* of these per-part memberships. Assuming that the uncertainties of the regions are independent, we show that the intersection of per-part RCIS is *equivalent* to the full-system RCIS. Thus, the compositional monitor is both *sound* and *complete*. Its complexity scales linearly with the number of parts. We demonstrate the effective application of the proposed monitor in a case study with a four-wheeled differential-drive robot persistently surveying a labyrinth.

The contributions of the paper are as follows:

i. Formalize persistent surveillance with black-box control as a hybrid system whose uncertainties evolve via bounded LPV factors.
ii. Compute per-part RCIS and prove their *soundness* and *completeness* with respect to the full RCIS.
iii. Propose a monitor based on the compositional RCIS with polyhedral modes and constant-time online membership checks suitable for embedded deployment.
iv. Demonstrate the effectiveness of the approach to monitor persistent surveillance of a labyrinth by a real robot.

The remainder of the paper is structured as follows. Section II discusses related work. Section III states the monitoring problem for persistent surveillance and introduces a running example used throughout the paper. Section IV presents the monitoring approach based on formalizing the hybrid model, obtaining invariants, and showing how to use them for monitoring. Section V evaluates the approach in a laboratory case study. Section VI discusses assumptions, limitations, and extensions; Section VII concludes the paper.

II. Related Work

A large body of work focuses on patrols over partitioned environments, ranging from waypoint/visitation sequencing [8], [9] to hybrid planners that account for region-dependent dynamics [6], as well as multi-robot coordination [10]. These approaches output trajectories or visitation schedules and typically provide completeness or optimality guarantees under the modeled dynamics and costs, assuming access to (or direct control over) the planning and control interfaces.

V. Nenchev and P. Sotiriadis are with Department of Electrical and Computer Engineering, University of the Bundeswehr Munich, 85579 Neubiberg, Germany. {vladislav.nenchev, prodromos.sotiriadis}@unibw.de

Necessary and sufficient conditions on robot speed controllers along a predefined path were derived for keeping the accumulation function for persistent surveillance bounded [11]. Prior work has also addressed autonomy under formal specifications [12] and an event-triggered task structure [13], in which policies are computed from modeled dynamics under uncertainty. Our runtime monitoring is complementary to all of these approaches: a planner can be wrapped by our monitor to guard against stalls, sensor faults, or adversarial perturbations that invalidate planning assumptions.

Runtime Verification (RV) frameworks monitor executions against formal specifications with strong correctness guarantees [14]. Temporal logic enables quantitative monitoring of real-valued traces [15], [16]. Although RV offers rigorous semantics and mature tooling, monitoring rich specifications can be computationally heavy onboard resource-constrained robots, and many approaches assume gray-/white-box access to the system. Progress and failure monitors based on stability certificates and short-horizon rollouts [17] were proposed for goal-directed navigation, which, however, are not easily extendable for persistent surveillance.

Hamilton-Jacobi reachability can also be used to provide monitors with safety guarantees via value functions over the state space [18]. The resulting certificates reduce runtime checks to set membership, but memory and computation requirements suffer from the curse of dimensionality, thus, limiting scalability for robotic systems. A Control Barrier Function (CBF) may enforce forward invariance via solving a Quadratic Programming (QP) online, offering safety guarantees with minimal intervention [19]. Model predictive control yields constraint satisfaction and robustness through tube/robust formulations [20]. They require either solving optimization problems at runtime or access to the control loop, which may be incompatible with black-box autonomy.

Compared to logic- or optimization-heavy monitors, our method relies on precomputed RCIS sets and set-membership checks online, while retaining a provably sound and complete compositional construction that scales to larger environments.

III. Problem Statement

Let a robot with pose $x_k \in X$ at discrete time $k \in \mathbb{N}$ and a bounded velocity $u_k \in U \subset \mathbb{R}^2$ move according to

$$x_{k+1} = x_k + u_k T_s, \tag{1}$$

where T_s is the sampling time. The autonomy stack that generates u_k is treated as a *black box*.

For the persistent surveillance setup, consider a bounded planar workspace $X \subset \mathbb{R}^2$ partitioned into finitely many parts $P = \{P_1, \dots, P_N\}, P_i \subset X$ with pairwise disjoint interiors; parts may meet only on their boundaries (e.g., along shared edges or vertices). For each part P_i, let $\mathrm{Obs}_i \subset X$ denote the set of robot poses from which P_i is observed. Note that Obs_i may also consist of multiple (disconnected) parts. The uncertainty of part P_i is denoted by a scalar $z_{i,k} \in \mathbb{R}_+$ at time k. Uncertainty *decreases* while P_i is effectively observed, i.e., when $x \in \mathrm{Obs}_i$, and *increases* otherwise. For clarity of exposition, we assume an obstacle-free workspace, but we show how the presence of obstacles can be treated in the case study.

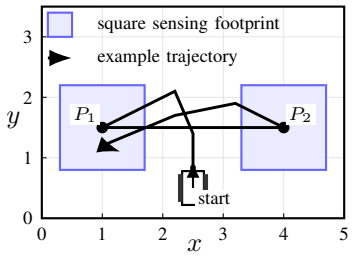


Fig. 1: Example: A robot persistently surveying two parts P_1, P_2 with a sample patrol trajectory.

Assumptions: (i) observability sets for each part are known or conservatively over-approximated; (ii) each $z_{i,k}$ evolves independently of $z_{j\neq i}$ given x_k (no cross-coupling among parts beyond the shared pose); (iii) uncertainty updates are bounded and monotone with observation status, e.g., multiplicative LPV factors with decay when observed and growth when unobserved; (iv) disturbances/parametric variations (including benign faults or adversarial perturbations within specification) are captured by these bounds; (v) at runtime, the monitor receives x_k; it may also receive u_k (optional) and either measured $z_{i,k}$ or computable proxies derived from sensing/coverage logs.

Problem 1: Given the available signals, design a *runtime monitor* that determines whether the persistent surveillance is being met, i.e., all parts remain within acceptable uncertainty, $z_{i,k} \in [0, z_i^{\mathrm{crit}}]$ for desired thresholds $z_i^{\mathrm{crit}} > 0$, and that raises (anticipatory) *alerts* when a violation is imminent. The monitor should run at predictable, low per-step cost suitable for onboard deployment, provide per-part diagnostics (which regions are at risk and by how much), scale to large N, and remain robust to bounded disturbances, parameter drift, and adversarial perturbations that are consistent with the specified observation model.

Example. We instantiate $X = [0, 5] \times [0, 3.5]$ with two parts centered at $p_1 = (1, 1.5)$ and $p_2 = (4, 1.5)$. Each part has a *square sensing footprint* of half-side $r = 0.7$, i.e., $\mathsf{obs}_i(x) = 1 \iff \|x - p_i\|_\infty \le r$, so the observation regions are polyhedral and used *as is* in computation. The sampling time is $T_s = 1\,\mathrm{s}$ and the robot follows the single-integrator model $x_{k+1} = x_k + u_k T_s$ with input bound $\|u_k\|_\infty \le 0.6$ (i.e., $U = \{u : \|u\|_\infty \le 0.6\}$). Uncertainties have to satisfy $0 \le z_i \le z^{\mathrm{crit}} = 4$. Observed/unobserved multiplicative factors lie in intervals $a \in [0.65, 0.80]$ and $b \in [1.05, 1.15]$; for robustness we use the monotone worst-case vertex $(a_{\max}, b_{\max}) = (0.80, 1.15)$ in each region. This example shown in Figure 1 will illustrate the proposed solution throughout the paper.

IV. Invariant-Based Monitoring

This section presents the key components of the approach as shown in Fig. 2: formalizing the hybrid model for persistent surveillance, introducing the RCIS and a compositional

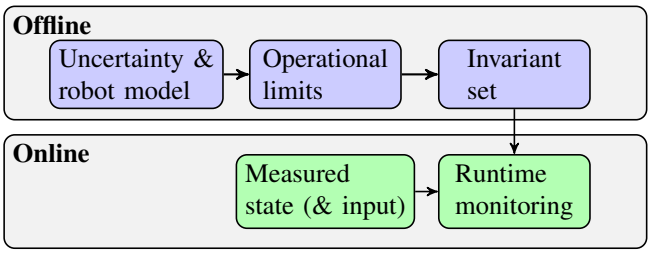


Fig. 2: Overview of the monitoring approach: In the *Offline* stage (top) the certificate is synthesized from the robot motion model and operational limits; in the *Online* stage (bottom) the measured state (and, optionally, the input) are used to check if operation remains in the invariant.

construction with soundness/completeness guarantees, computing them, and using invariants as monitoring primitives.

A. Hybrid model

We model the closed loop as a state-dependent hybrid system that contains the motion of the robot (1) and multiplicative, region-dependent uncertainty dynamics:

$$z_{i,k+1} = \gamma_i(x_k)\, z_{i,k}, \qquad i = 1, \dots, N, \tag{2}$$

where $\gamma_i(x)$ is changes according to the robot's location:

$$\gamma_i(x) \in \begin{cases} [a_i^{\min}, a_i^{\max}] \subset (0,1), & x \in \mathrm{Obs}_i, \\ [b_i^{\min}, b_i^{\max}] \subset (1,\infty), & \text{else.} \end{cases}$$

For each part i, we build a PWA system in the state $\xi_i = (x, z_i, u)$ with two modes:

$$\begin{aligned} \mathrm{obs}_i &: x^+ = x + uT_s, \quad z_i^+ = a_i^{\max} z_i, \quad x \in \mathrm{Obs}_i; \\ \mathrm{unobs}_i &: x^+ = x + uT_s, \quad z_i^+ = b_i^{\max} z_i, \quad x \in X \setminus \mathrm{Obs}_i, \end{aligned}$$

subject to the joint domain $D_i := X \times [0, z_i^{\mathrm{crit}}] \times U$. Further, define guards $G_i^{\mathrm{obs}} := \{x \in X \mid x \in \mathrm{Obs}_i\}$ and $G_i^{\mathrm{ubobs}} := \{x \in X \mid x \notin \mathrm{Obs}_i\}$.

Example (revisited). In our running example, for each $i \in \{1, 2\}$ we build a PWA subsystem in $[x^\top,\, z_i]^\top \in \mathbb{R}^3$ with input $u \in \mathbb{R}^2$ and two mode maps:

$$\mathrm{obs}_i : \begin{cases} x^+ = x + uT_s, \\ z_i^+ = a_{\max}\, z_i, \end{cases} \qquad \mathrm{unobs}_i : \begin{cases} x^+ = x + uT_s, \\ z_i^+ = b_{\max}\, z_i, \end{cases}$$

guarded by $x \in \mathrm{Obs}_i := \{\, \|x - p_i\|_\infty \le r \,\}$ and $x \in X \setminus \mathrm{Obs}_i$, respectively. Because $X \setminus \mathrm{Obs}_i$ is non-convex, it is represented by a union of polyhedral sets with a separate linear-time-invariant-mode per set. To reduce the complexity of the example, we consider only the mode between the two to-be-observed parts, and the overall hybrid system consists of three modes.

B. Invariants

Let $\xi_k := [\, x_k^\top,\ [z_{1,k}, \dots, z_{N,k}],\ u_k^\top \,]^\top$ denote the joint state–input vector with domain

$$D := X \times \prod_{i=1}^{N} [0, z_i^{\mathrm{crit}}] \times U.$$

A set $\mathcal{S}$ is RCIS for (1)–(2) if, for all admissible $u_k \in U$ and for all factors $\gamma_i(\cdot)$ consistent with the active region at x_k, $\xi_k \in \mathcal{S} \;\Rightarrow\; \xi_{k+1} \in \mathcal{S}$.

Computing the maximal full-system RCIS $\mathcal{S}$ scales poorly with the number of parts N, making exact computation and storage quickly prohibitive. Thus, we compute per-part RCIS's and intersect them to retain correctness. For each part i, consider the subsystem over the state space (x, z_i), dynamics (1) with the input set U and (2) only for z_i, and the corresponding region guards. Then, compute its maximal RCIS $\mathcal{S}_i^{\max}$ in $D_i := X \times [0, z_i^{\mathrm{crit}}] \times U$. For multiplicative updates $z_i^+ = \gamma_i z_i$ with γ_i bounded per region, the next-step map is monotone in γ_i. Hence, the RCIS set coincides with the set computed at the region-wise *worst-case* vertices $(a_i^{\max}, b_i^{\max})$ (equivalently, intersect vertex-wise invariants if multiple vertices are used). For a set $S_i \subseteq D_i$, define the one–step predecessors in mode i as

$$\mathrm{Pre}_i(S_i) := \big\{\, \hat{x} \in D_i :\ f_i(\hat{x}) \in S_i,\ \forall u \in U,\ \forall \gamma_i \in \Gamma_i(x) \,\big\}.$$

This leads to the following proposition.

Proposition 1: Assume:

(i) $\forall i$, z_i^+ in (2) depends only on z_i and the guard Obs_i, and not on z_j with $j \neq i$;

(ii) The full system and each per-part subsystem use the same workspace set X and input set U;

(iii) The full system and each per-part subsystem use the same guards G_i^{obs}, G_i^{unobs} and uncertainty/LPV bounds.

Let $\mathcal{S} \subseteq D$ denote the maximal RCIS set for the full system, and let $\mathcal{S}_i^{\max} \subseteq D_i$ denote the maximal RCIS set for the ith per-part subsystem. Then, with $i = 1, \dots, N$ and

$$\hat{\mathcal{S}} \;:=\; \big\{ [x^\top, z^\top, u^\top]^\top \in D \,\big|\, [x^\top, z_i, u^\top]^\top \in \mathcal{S}_i^{\max} \quad \forall i \big\},$$

$\mathcal{S} = \hat{\mathcal{S}}$ holds.

Proof: We show both inclusions.

1) $\mathcal{S} \subseteq \hat{\mathcal{S}}$. Fix $i \in \{1, \dots, N\}$ and consider the projection

$$\Pi_i : D \to D_i, \qquad \Pi_i(x, z, u) := (x, z_i, u).$$

Let $P_i := \Pi_i(\mathcal{S}) \subseteq D_i$. We claim that P_i is RCIS for the ith subsystem. Take any $(\bar{x}, \bar{z}_i, \bar{u}) \in P_i$. By definition, there exists $\bar{z}_{-i}$ such that $(\bar{x}, \bar{z}, \bar{u}) \in \mathcal{S}$. Since $\mathcal{S}$ is RCIS for the full system, for any admissible uncertainty realization (equivalently, any $\gamma(\cdot) \in \Gamma(\bar{x})$) and the induced successor

$$\bar{x}^+ = \bar{x} + \bar{u}T_s, \qquad \bar{z}^+ = g(\bar{x}, \bar{z}),$$

we have $(\bar{x}^+, \bar{z}^+, \tilde{u}) \in \mathcal{S}$ for all $\tilde{u} \in U$ (as required by the predecessor definition used in the paper). By Assumption (i), $\bar{z}_i^+$ depends only on $(\bar{x}, \bar{z}_i)$ (and the guard Obs_i), hence $\bar{z}_i^+ = g_i(\bar{x}, \bar{z}_i)$ is exactly the successor of the ith subsystem under the same guard and uncertainty bounds (Assumption (iii)). Projecting yields $(\bar{x}^+, \bar{z}_i^+, \tilde{u}) \in P_i$ for all $\tilde{u} \in U$ and all admissible uncertainties, i.e., P_i is RCIS for the ith subsystem. By maximality of $\mathcal{S}_i^{\max}$ on D_i, we obtain $P_i \subseteq \mathcal{S}_i^{\max}$, hence

$$(x, z, u) \in \mathcal{S} \;\Rightarrow\; (x, z_i, u) \in \mathcal{S}_i^{\max} \quad \forall i \;\Rightarrow\; (x, z, u) \in \hat{\mathcal{S}}.$$

Therefore, $\mathcal{S} \subseteq \hat{\mathcal{S}}$.

2) $\hat{\mathcal{S}} \subseteq \mathcal{S}$. We first show that $\hat{\mathcal{S}}$ is RCIS for the full system. Take any $(\bar{x}, \bar{z}, \bar{u}) \in \hat{\mathcal{S}}$. By definition, $(\bar{x}, \bar{z}_i, \bar{u}) \in \mathcal{S}_i^{\max}$ for all i. Let an admissible uncertainty realization be fixed (equivalently, choose any $\gamma(\cdot) \in \Gamma(\bar{x})$), with successor

$$\bar{x}^+ = \bar{x} + \bar{u}T_s, \qquad \bar{z}_i^+ = g_i(\bar{x}, \bar{z}_i), \quad i = 1, \ldots, N.$$

Because each $\mathcal{S}_i^{\max}$ is RCIS for the corresponding per-part subsystem and, by Assumptions (ii)–(iii), uses the same X, U, guards, and uncertainty bounds as the full system,

$$(\bar{x}^+, \bar{z}_i^+, \tilde{u}) \in \mathcal{S}_i^{\max} \qquad \forall i, \ \forall \tilde{u} \in U.$$

Thus, $(\bar{x}^+, \bar{z}^+, \tilde{u}) \in \hat{\mathcal{S}}$ for all $\tilde{u} \in U$ and all admissible uncertainties, i.e., $\hat{\mathcal{S}}$ is RCIS for the full system and satisfies $\hat{\mathcal{S}} \subseteq D$. By maximality of $\mathcal{S}$ among RCIS subsets of D, we conclude $\hat{\mathcal{S}} \subseteq \mathcal{S}$.

Combining 1) and 2) gives $\mathcal{S} = \hat{\mathcal{S}}$. ∎

Remark 1: Proposition 1 implies that the compositional invariant is *sound* (no false alarms) and *complete* (no misses) w.r.t. $\mathcal{S} \subseteq D$. Under static, known obstacles, the obstacle-free portion of X can be under-approximated by a finite collection of convex polyhedral cells; this under-approximation preserves soundness (see case study). Dynamic obstacles are not considered in this work.

C. Computing invariant sets

The RCIS sets are obtained as *greatest fixed points* of the monotone set operator $T(S) = D \cap \mathrm{Pre}(S)$, iterated from $S^{(0)} = D$ until convergence [7]. For linear (or modewise-affine) dynamics with polyhedral constraints, $\mathrm{Pre}(\cdot)$ reduces to polyhedral operations (linear images, Minkowski sums, Pontryagin differences) and Boolean combinations, yielding polyhedral RCIS sets (exact or convergent approximations). For the LPV states (2), we use the standard *vertex method*: compute (or intersect) invariants for each vertex system $(a_i^{\max}, b_i^{\max})$ consistent with the active region, which is conservative and monotone with respect to multiplicative uncertainty [7], [21]. We encode (1)–(2) as a hybrid LPV/PWA model with polyhedral constraints and compute the RCIS's via fixed-point iteration. To obtain the compositional invariants, we iterate the robust predecessor operator per mode under its guard and intersect across modes until a fixed point is reached; the result is a union of polytopes $\mathcal{S}_i^{\max} \subset D_i$ (Algorithm 1).

Example (revisited). We compute two per-part invariants in $[x, y, \ z_i, \ v_x, v_y]$ (two modes per part: observed/unobserved) and, for comparison, a single full invariant in $[x, y, \ z_1, z_2, \ v_x, v_y]$ (three modes: observing P_1, observing P_2, observing none). We set $(a_{\max}, b_{\max}) = (0.80, 1.15)$ in the observed/unobserved regions. Because the parts are denoted by squares, there is no geometric approximation gap. Table I compares the complexity of the invariants in terms of state dimension, number of system modes, invariant polytopes and vertices, and computation time on a general purpose laptop.

Algorithm 1 Per-part compositional invariant (offline)

Require: X, U, centers p_i, half-side S, bounds z_i^{crit}, worst-case $(a_i^{\max}, b_i^{\max})$, T_s, tolerance ε, max iters $L_{\max}$

Define guards: $\mathrm{Obs}_i = \{x : \|x - p_i\|_\infty \le S\}$, $\mathrm{Unobs}_i = X \setminus \mathrm{Obs}_i$

Define mode maps on (x, z_i, u):
  Obs: $x^+ = x + uT_s$, $z_i^+ = a_i^{\max} z_i$; Unobs: $x^+ = x + uT_s$, $z_i^+ = b_i^{\max} z_i$

Initialize $S^{(0)} := D_i := X \times [0, z_i^{\mathrm{crit}}] \times U$

for $\ell = 0, 1, \ldots, L_{\max}$ **do**

  $\widehat{S}^{(\ell+1)} := \mathrm{Pre}_{\mathrm{Obs},i}\big(S^{(\ell)}\big) \ \cap \ \mathrm{Pre}_{\mathrm{Unobs},i}\big(S^{(\ell)}\big)$

  $S^{(\ell+1)} := D_i \cap \widehat{S}^{(\ell+1)}$ ▷ intersect with joint box

  if $S^{(\ell+1)} = S^{(\ell)}$ (or $\mathrm{dist}_H < \varepsilon$) **then break**

  end if

end for

$\mathcal{S}_i^{\max} \leftarrow$ minimal union-of-polytopes representation of $S^{(\ell+1)}$ (remove redundant facets/sets)

return $\mathcal{S}_i^{\max}$

TABLE I: Running example: invariant complexity.

System	State Dim.	# modes	# P	# Vertices	Comp. time [s]
Part 1	3	2	12	72	2.4
Part 2	3	2	18	108	3.4
Full	4	3	610	9280	3319

D. Monitoring using invariant sets

Using full invariant: At time k, the system (1)–(2) is healthy iff $\xi_k \in \mathcal{S}$. Leaving $\mathcal{S}$ flags a failure or, if u_k is also available and the resulting $\xi_{k+1} \notin \mathcal{S}$ based on (1) – an imminent failure.

Using compositional invariants: At time k, form $\xi_{i,k} = [x_k^\top, \ z_{i,k}, \ u_k^\top]^\top$ (or drop u_k and encode its bound in the offline construction) and test membership in $\mathcal{S}_i^{\max}$ for *all* i. Following Proposition 1, the compositional decision is

$$\text{healthy at } k \iff (x_k, z_{i,k}, u_k) \in \mathcal{S}_i^{\max} \ \forall i \quad (\text{i.e., } \xi_k \in \hat{\mathcal{S}}).$$

Membership reduces to halfspace checks against the facets of the polytopes in the union; for the compositional monitor the per–step cost scales linearly with N.

Example (revisited). For a measured triple $(x_k, z_{1,k}, z_{2,k})$ (and optional u_k) the system is declared *healthy* iff

$$(x_k, z_{1,k}, u_k) \in \mathcal{S}_1^{\max} \quad \text{and} \quad (x_k, z_{2,k}, u_k) \in \mathcal{S}_2^{\max}.$$

The full invariant $\mathcal{S}$ can also be used for monitoring analogously. On our platform on average, the compositional check ran in 0.03 ms per step, while the check with the full invariant took 1 ms per step without any numerical optimizations.

V. Case Study

We evaluate the proposed invariant-based monitor on a four-wheeled differential-drive robot tasked with persistently surveying a labyrinth-like indoor environment. The study follows the workflow (Figure 2) introduced earlier: per-part robust invariants are synthesized offline from a LPV-hybrid model and operational limits; the monitor performs

Algorithm 2 Compositional monitoring (online)

Require: $\{\mathcal{S}_i^{\max}\}_{i=1}^N$, measurements $(x_k, \{z_{i,k}\})$, optional u_k

1: **for** $i = 1$ to N **do**
2: $\quad h_i \leftarrow \mathbf{1}\{(x_k, z_{i,k}, u_k) \in \mathcal{S}_i^{\max}\}$ ▷ polyhedral union membership
3: **end for**
4: **if** $\prod_{i=1}^N h_i = 1$ **then return** Healthy
5: **else return** Alert (report indices $\{i : h_i = 0\}$ and nearest-facet margins)

set-membership checks to raise anticipatory alerts. We only evaluate the *compositional* invariant monitor, as computing the *full* invariant is intractable.

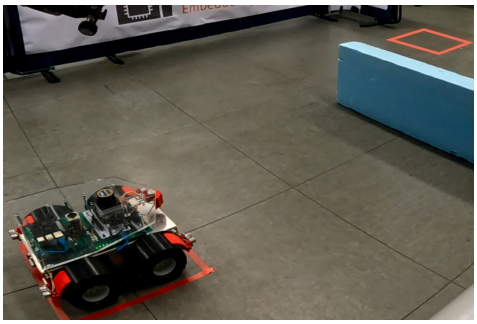
A. Experimental Setup

Robot: The platform is a ground robot (Figure 3a) with footprint $0.44 \times 0.40\,\text{m}$ (mass 5 kg), wheel encoders (16 pulses/rev), an ITG-3200 gyroscope (10 Hz), and a Hokuyo URG-04LX 2D LiDAR (10 Hz, 240° FOV). Low-level control runs on an STM32F746ZG (Cortex-M7, 1 kHz), while high-level software runs on a Raspberry Pi 4B (8 GB) with ROS 2; actuator bounds are $v \in [-0.26, 0.26]$ m/s, $\omega \in [-1, 1]$ rad/s and the state space is $x, y \in [-4, 4] \times [-3, 3]$ m, $\theta \in [-\pi, \pi]$, with sampling time $T_s = 0.1$ s.

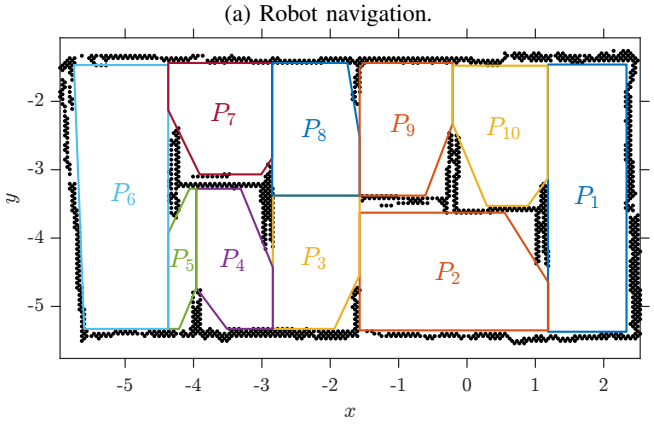
Policy: We use a simple uncertainty–greedy patrol policy: the robot selects the partition with the largest current uncertainty z_i, plans a path to the center of that partition, and dwells until z_i drops below a target level z_i^{tar}; it then reselects the next part. A small hysteresis band on z_i and a minimum inter-visit time $\tau_{\min}$ prevent chattering. Local motion is handled by ROS Nav2.

Environment: The traversable obstacle-free labyrinth workspace is manually under-approximated by a finite number of partially adjacent convex polyhedral cells that serve as observability regions and mode domains in the hybrid model. The resulting map and labels are shown in Figure 3b. Each partition P_i is equipped with an uncertainty proxy z_i that evolves multiplicatively according to the LPV model with part–specific factors (a_i, b_i), following $z_{i,k+1} = \gamma_i(x_k)\, z_{i,k}$, with $\gamma_i(x) \in [a_i^{\max}, 1)$ when $x \in P_i$ (observed) and $\gamma_i(x) \in (1, b_i^{\max}]$ otherwise. $z_{max} = 10$ for all parts.

Invariant set computation: We conservatively instantiate the input set as $U = \{u : \|u\|_\infty \le 0.26\}$ obtained from platform bounds, with an extra inflation $\delta_u = 0.1$ to cover Nav2 tracking error. This over-approximates the instantaneous body-frame linear velocity projected in the world frame; heading variation within one sample is absorbed by universal quantification over $u \in U$. For each part i we build a two-mode PWA subsystem in (x, z_i, u) with guards $x \in P_i$ vs. $x \notin P_i$ and compute the maximal RCIS set $S_i^{\max} \subset X \times [0, z_i^{\text{crit}}] \times U$ via fixed-point iteration of the predecessor operator under universal quantification over inputs and LPV vertices (Algorithm 1). Computation took between 111 s and 154 s for each compositional invariant on a general-purpose laptop.



(a) Robot navigation.



(b) Labyrinth configuration.

Fig. 3: Labyrinth for persistent surveillance by a four-wheeled differential-drive robot. The labyrinth has been manually partitioned into 10 parts.

B. Monitoring

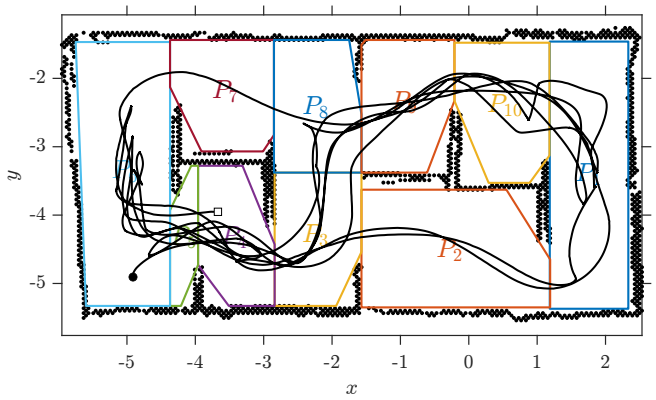
From each run we obtain time-stamped robot poses $x_k \in \mathbb{R}^2$ (filtered using a Kalman filter) and per-part uncertainty proxies $z_{i,k}$ (derived from perception/coverage events). We derive the uncertainty factors (a_i, b_i) online from LiDAR measurements as follows.

Uncertainty update: We have excluded the LiDAR detections of the initial mapping pass shown in Figure 3b when defining the partitions. To emulate time-varying uncertainty, additional small objects were placed in the labyrinth during operation without preventing the robot from moving freely through all parts. New detections are accumulated per partition in a counter $C_i(k) > 0$ at time k (detections since the last time the robot entered P_i). When the robot *re-enters* P_i at time k, we map $r_i(k) := C_i(k)/C_i(k-1)$ monotonically into the decay range $[a_i^{\min}, a_i^{\max}]$ by

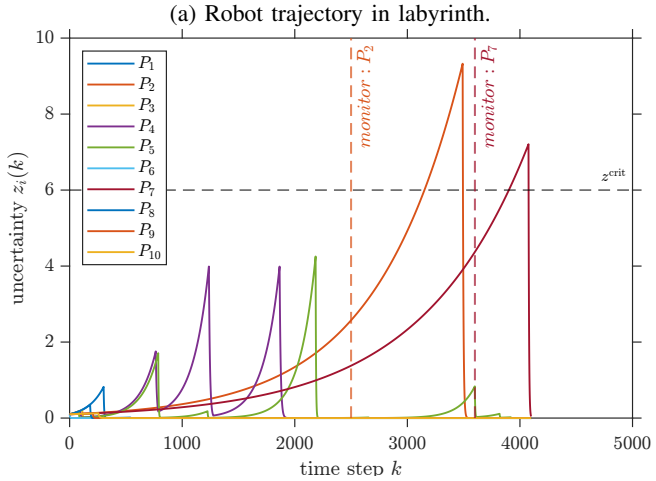
$$a_i(k_{\text{in}}) = \text{clip}\Big(a_i^{\min} + \frac{\min\{r_i,\ r_i^{\max}\} - 1}{r_i^{\max} - 1}\,(a_i^{\max} - a_i^{\min})\Big).$$

Here $r_i^{\max} > 1$ is a chosen saturation point and $\text{clip}(\cdot)$ enforces the interval $[a_i^{\min}, a_i^{\max}]$. Intuitively, more detections drive a_i upward toward $a_i^{\max} < 1$, meaning uncertainty decays more slowly even when the partition is observed.

For simplicity in this case study, the unobserved growth factor b_i was kept constant, i.e., $b_i \in (1, b_i^{\max}]$ independent of



(a) Robot trajectory in labyrinth.



(b) Uncertainty evolution and monitor triggers.

Fig. 4: Labyrinth for persistent surveillance by a four-wheeled differential-drive robot. The labyrinth has been manually partitioned into 10 parts.

r_i. We did not investigate removing obstacles; thus, C_i^{new} is non-decreasing between entries and a_i is piecewise-constant and non-decreasing across visits. The parameters are updated while moving through P_i. The same z_i sequences are then fed to the monitor according to Algorithm 2.

Evaluation: We monitor a run over multiple minutes in the labyrinth with a failing patrol that loiters and starves two parts, causing uncertainty growth beyond admissible bounds. For reference, we compare against a threshold-only rule that raises an alert when any $z_{i,k} \geq z_i^{\text{crit}}$. Further, we evaluate: (i) *Detection lead time* from the monitor's first alert to the threshold crossing $z_{i,k} \geq z_i^{\text{crit}}$ (positive values indicate anticipatory detection); (ii) *false alerts* (alerts without subsequent violation within a grace window); and (iii) checking computation time.

C. Results

Figure 4a shows the robot position trajectory overlaid on the ten-part partition; Figure 4b plots the corresponding uncertainty traces and marks monitor triggers. The invariant-based monitor raises alerts *before* uncertainties hit the critical threshold, providing anticipatory warnings; the threshold-only baseline, by definition, reacts at (or after) violation. No false alerts are produced by the monitor.

During the run patrolling fails, loitering in a subset of the maze leads to starvation of two parts; the monitor exits the invariant set well before z_i crosses z_i^{crit}, yielding positive lead time. Subsequent growth of z_i confirms the early warning, and the parts reported by the monitor match the regions visibly under-serviced in the trajectory plot.

We ran three independent trials per configuration with different random seeds for patrol initialization and obstacle placements. Each of them yielded similar results in terms of monitoring outcomes. On average, per-step runtime for checking polyhedral union membership over all parts was 0.3 ms; storing per-part unions and their facet matrices dominates memory consumption. We plan to release the scripts for generating the experimental evaluation.

VI. Discussion

This section reflects on assumptions, practical trade-offs in using the proposed monitor, and directions for extensions.

Guarantees and assumptions: The completeness and soundness of the monitor based on compositional RCIS relies on two conditions: (i) mutual *independence* of the part-wise uncertainties, and (ii) *robust* (universal) quantification over the same state and input space, guards, and LPV bounds. Condition (i) is appropriate when each uncertainty state is a local proxy driven by sensing staleness/coverage (e.g., time-since-last-observation or map freshness). However, global events may couple the uncertainty of parts. For example, wind-driven smoke propagation in fire monitoring can raise uncertainty beyond one part. In this case, one can cluster coupled parts into multi-part subsystems, albeit at the cost of higher RCIS complexity. In the case study, the under-approximation of the traversable space yields *conservative* invariants. Thus, within the under-approximated area runtime alerts remain *sound* but can be *conservative*.

Measurement noise and model mismatch: With noisy x_k or $z_{i,k}$, sharp set boundaries can induce chattering. Two possible standard remedies are (i) *guard erosion*/set inflation: shrink Obs_i by a margin δ or inflate the invariant to accommodate estimation error; and (ii) *hysteresis*: declare exit only after m consecutive violations or require a margin $\eta > 0$ beyond the boundary. If u_k is not available, quantify over U offline; if it is available, slice the union by the measured u_k to reduce conservatism.

Complexity: The offline computation of the RCIS's is based on a fixed-point iteration of robust predecessor operators with polyhedral set operations. In the worst case, both runtime and representation size grow exponentially with the state dimension and with the number of halfspaces/vertices. In our compositional construction, this cost is incurred in the dimension of $[x^\top, z_i]^\top$, whereas the full-system RCIS requires the same operations in $[x^\top, z_1, \ldots, z_N]^\top$. As N increases, computation and storage become quickly prohibitive, which is reflected in Table I, where the full invariant has orders-of-magnitude more vertices and higher runtime.

Online runtime scales with the number of polytopes and facets. For compositional monitoring, this scales roughly linearly with N (each membership test is independent), and unions can be compressed via polytope merging, facet pruning, or spatial indexing (e.g., k-d trees on polytope bounding boxes). Upon memory limitations, the union can be approximated by a single outer polytope around each $\mathcal{S}_i^{\max}$ and the approximation gap can be tracked.

Choice of thresholds: The thresholds z_i^{crit} encode the surveillance objective and reflect sensing fidelity, revisit constraints, and acceptable risk. Unequal priorities are handled by part-specific z_i^{crit} and by weighting the reporting (e.g., alert sooner on critical parts using larger safety margins).

Observability and motion models: Both the running example and the case study used polyhedral representations of the environment. For circular cones, occlusions, or complex visibility polygons, inner/outer polyhedral approximations to trade tightness for tractability can be used. The single-integrator kinematics can be replaced by other linear (or modewise linearized) models. The invariant computation proceeds analogously, albeit in a higher-dimensional space and with possibly significantly higher computational complexity.

Adversarial robustness and cyber–physical attacks: Because the monitor relies on measured state, integrity attacks that spoof these channels can mask violations. Practical mitigation includes cross-checks between commanded u_k and observed motion (residual tests), redundant sensing for $z_{i,k}$ (e.g., overlapping footprints), and secure logging/attestation for the signals used by the monitor. From a synthesis viewpoint, suspicious channels can be treated as adversarial disturbances by enlarging the LPV bounds, which preserves soundness at the cost of more conservative alerts.

Limitations: Exact computation of RCIS in high dimensions remains challenging; our compositional construction mitigates, but cannot remove, the intrinsic cost for very large N without sacrificing some tightness.

Extensions: The framework extends to 3D workspaces, to multi-robot surveillance (modularity remains sound if interactions are conservatively bounded), and to stochastic envelopes (replace worst-case vertices by chance-constrained predecessors to trade false positives for false negatives). Hybridization with other monitoring objectives is another natural extension, e.g., combining with barrier-methods-based monitors for richer early-warning signals, or adding look-ahead for imminent failures.

VII. Conclusion

We introduced a runtime monitoring framework for persistent surveillance that treats the autonomy stack as a black box. The key idea is a compositional construction of invariant sets for each to-be-surveyed part and checking their conjunction online. Under independence and common-domain assumptions, we showed that the compositional monitor is equivalent to the full-system-based monitor. The monitor requires only constant-time polyhedral membership tests online and scales linearly with the number of parts. A laboratory case study with a mobile robot in a labyrinth illustrated how the construction is instantiated from data and how impending instability can be detected early.

Future work will include incorporating coupling or disturbances in per-part predecessors and extending to multi-robot teams with collision and communication constraints.

References

[1] R. N. Smith, M. Schwager, S. L. Smith, B. H. Jones, D. Rus, and G. S. Sukhatme, "Persistent ocean monitoring with underwater gliders: Adapting sampling resolution," *Journal of Field Robotics*, vol. 28, no. 5, pp. 714–741, 2011.

[2] D. W. Casbeer, D. B. Kingston, R. W. Beard, and T. W. McLain, "Cooperative forest fire surveillance using a team of small unmanned air vehicles," *Int. J. Syst. Sci.*, vol. 37, no. 6, pp. 351–360, 2006.

[3] X. Lan and M. Schwager, "Planning periodic persistent monitoring trajectories for sensing robots in Gaussian Random Fields," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2013, pp. 2415–2420.

[4] J. Chen, T. Shu, T. Li, and C. W. de Silva, "Deep Reinforced Learning Tree for Spatiotemporal Monitoring With Mobile Robotic Wireless Sensor Networks," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 11, pp. 4197–4211, 2020.

[5] J.-S. Ha and H.-L. Choi, "On periodic optimal solutions of persistent sensor planning for continuous-time linear systems," *Automatica*, vol. 99, pp. 138–148, 2019.

[6] J. Hall, C. G. Cassandras, and S. B. Andersson, "Persistent Monitoring Trajectory Optimization in Partitioned Environments," in *American Control Conf. (ACC)*, 2025, pp. 3813–3818.

[7] F. Blanchini, "Set invariance in control," *Automatica*, vol. 35, no. 11, pp. 1747–1767, 1999.

[8] J. Yu, S. Karaman, and D. Rus, "Persistent Monitoring of Events With Stochastic Arrivals at Multiple Stations," *IEEE Transactions on Robotics*, vol. 31, no. 3, pp. 521–535, 2015.

[9] A. Jones, M. Schwager, and C. Belta, "Information-guided persistent monitoring under temporal logic constraints," in *2015 American Control Conference (ACC)*, 2015, pp. 1911–1916, iSSN: 2378-5861.

[10] N. Zhou, X. Yu, S. B. Andersson, and C. G. Cassandras, "Optimal Event-Driven Multiagent Persistent Monitoring of a Finite Set of Data Sources," *IEEE Trans. Autom. Control*, vol. 63, no. 12, pp. 4204–4217, 2018.

[11] S. L. Smith, M. Schwager, and D. Rus, "Persistent monitoring of changing environments using a robot with limited range sensing," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2011, pp. 5448–5455.

[12] V. Nenchev and C. Belta, "Receding Horizon Robot Control in Uncertain Environments with Temporal Logic Constraints," in *15th European Control Conf. (ECC)*, 2016, pp. 2614–2619.

[13] V. Nenchev, C. G. Cassandras, and J. Raisch, "Event-driven optimal control for a robotic exploration, pick-up and delivery problem," *Nonlinear Analysis: Hybrid Systems*, vol. 30, pp. 266–284, 2018.

[14] M. Leucker and C. Schallhart, "A brief account of runtime verification," *The Journal of Logic and Algebraic Programming*, vol. 78, no. 5, pp. 293–303, 2009.

[15] O. Maler and D. Nickovic, "Monitoring Temporal Properties of Continuous Signals," in *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems*, Y. Lakhnech and S. Yovine, Eds. Berlin, Heidelberg: Springer, 2004, pp. 152–166.

[16] G. E. Fainekos and G. J. Pappas, "Robustness of temporal logic specifications for continuous-time signals," *Theoretical Computer Science*, vol. 410, no. 42, pp. 4262–4291, 2009.

[17] V. Nenchev and P. Sotiriadis, "Monitoring Progress and Failure in Autonomous Robot Navigation: A Case Study," in *Runtime Verification*, B. Könighofer and H. Torfah, Eds. Cham: Springer Nature Switzerland, 2026, pp. 317–335.

[18] I. Mitchell, A. Bayen, and C. Tomlin, "A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games," *IEEE Trans. Autom. Control*, vol. 50, no. 7, pp. 947–957, 2005.

[19] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control Barrier Function Based Quadratic Programs for Safety Critical Systems," *IEEE Trans. Autom. Control*, vol. 62, no. 8, pp. 3861–3876, 2017.

[20] D. Q. Mayne, "Model predictive control: Recent developments and future promise," *Automatica*, vol. 50, no. 12, pp. 2967–2986, 2014.

[21] J. S. Shamma, "An Overview of LPV Systems," in *Control of Linear Parameter Varying Systems with Applications*, J. Mohammadpour and C. W. Scherer, Eds. Boston, MA: Springer US, 2012, pp. 3–26.